Weekly report

1 Done

1.1 Paper Reading

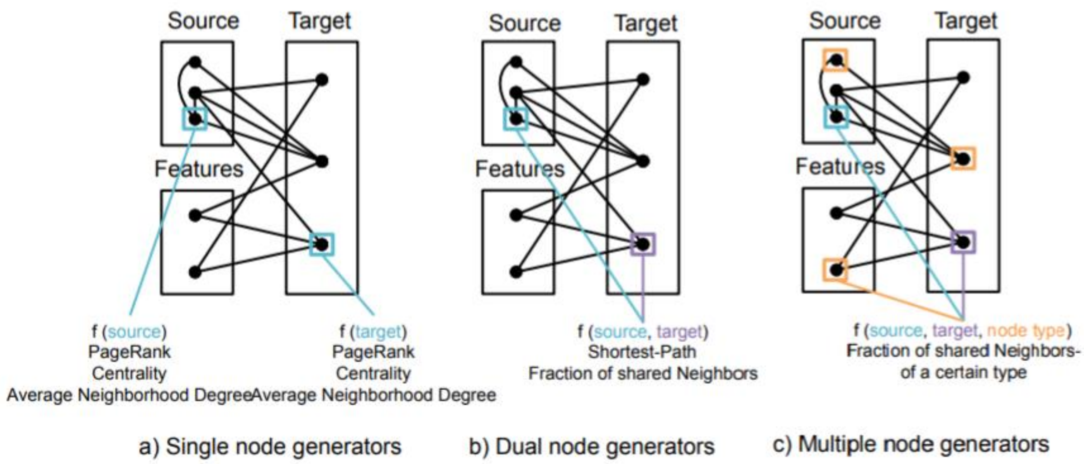
- *Graph Based Recommendations: From Data Representation to Feature Extraction and Application*

  This paper provides an approach to distill graph features by generators:

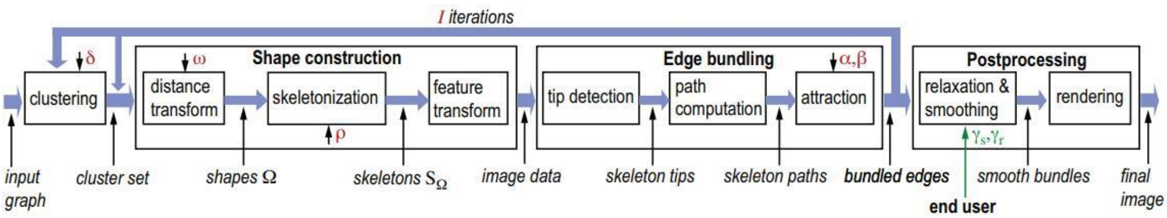
```
Algorithm 2: Extract graph features from a sub-graph

input : SubGraph - sub-graph derived from the complete graph representation
        PredEdge - edge type of the relationship being predicted
output: ExtractedSubGraphFeatures - set of features extracted from
        SubGraph

ExtractedSubGraphFeatures ← ∅
SubGraphPredictedEdges ← ExtractPredictedEdges(SubGraph,PredEdge)
foreach (SourceEntity,TargetEntity) of Edge ∈ SubGraphPredictedEdges
 do
    foreach 1-Function in 1-VertexGenerators do
        SourceFeatures ← 1-Function(SourceEntity)
        TargetFeatures ← 1-Function(TargetEntity)
        ExtractedSubGraphFeatures ← (ExtractedSubGraphFeatures ∪
         SourceFeatures ∪ TargetFeatures)
    end
    foreach 2-Function in 2-VertexGenerators do
        SourceTargetFeatures ← 2-Function(SourceEntity,TargetEntity)
        ExtractedSubGraphFeatures ← (ExtractedSubGraphFeatures ∪
         SourceTargetFeatures)
    end
    MultipleEntityCombinations ←
     ExtractEntityCombinations({VertexTypes})
    foreach EntityCombination ∈ MultipleEntityCombinations do
        N ← |EntityCombination|
        foreach N-Function in N-VertexGenerators do
            MultipleEntityFeatures ← N-Function(SourceEntity,
             TargetEntity, EntityCombination)
            ExtractedSubGraphFeatures ← (ExtractedSubGraphFeatures ∪
             MultipleEntityFeatures)
        end
    end
    return ExtractedSubGraphFeatures
end
```

The features here are divided into 1-Vertex, 2-Vertex and N-vertex.

1-vertex: Degree Centrality, Average Neighbor Degree, PageRank, Clustering Coefficient and Node Redundancy.
2-vertex and N-vertex: Shortest Path, Shared Neighbors of Type X and Complex relationships across entities.



- *Skeleton-based edge bundling for graph visualization*

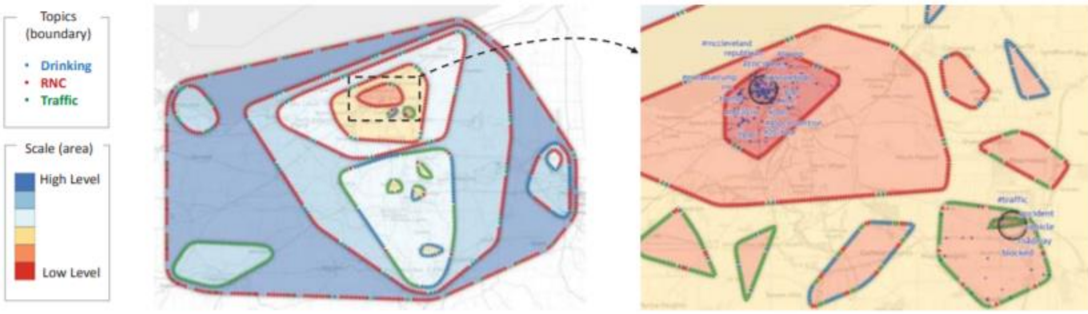


Edge bundling for skeletonization:

1) Cluster edges into groups, or clusters, C_i which have strong geometrical and optionally attribute-based similarity.
2) For each cluster, compute a thin shape surrounding its edges using a distance-based method.
3) For each shape, compute its skeleton and feature transform of skeleton.
4) For each cluster, attract its edges towards the skeleton using its feature transform of the skeleton.
5) Repeat the process form step1 or step 2 until the desired bundling level is reached.
6) Perform a final smoothing and next render the graph using a cushion-like technique to help understanding bundle overlaps.

- *TopoGroups: Context-Preserving Visual Illustration of Multi-Scale Spatial Aggregates*

Authors present a visual design for distribution patterns at multiple levels of scale.



The colorful dashed strokes encoding the category distribution in an area. I think that the length of the line segment can be further studied. Besides, authors mention that blue - yellow - red is a nice choice of color matching on the map.

- *Local Higher-Order Graph Clustering*

This work develops the Motif-based Approximate Personalized PageRank algorithm that finds clusters containing a seed node with minimal motif conductance.

Algorithm 1: Motif-PageRank-Nibble method for finding localized clusters with small motif conductance.

Input: Unweighted graph $G = (V, E)$, motif M, seed node u, teleportation parameter α, tolerance ε

Output: Motif-based cluster (set $S \subset V$)

$W_{ij} \leftarrow$ #(instances of M containing nodes i and j)

$\tilde{p} \leftarrow$ Approximate-Weighted-PPR$(W, u, \alpha, \varepsilon)$ (Algorithm 2)

$D_W \leftarrow \text{diag}(We)$

$\sigma_i \leftarrow$ ith smallest entry of $D_W^{-1}\tilde{p}$

return $S \leftarrow \arg\min_\ell \phi_M(S_\ell)$, where $S_\ell = \{\sigma_1, \ldots, \sigma_\ell\}$

Algorithm 2: Approximate-Weighted-PPR

Input: Undirected edge-weighted graph $G_w = (V_w, E_w, W)$, seed node u, teleportation parameter α, tolerance ε

Output: an ε-approximate weighted PPR vector $\tilde{p}$

$\tilde{p}(v) \leftarrow 0$ for all vertices v

$r(u) \leftarrow 1$ and $r(v) \leftarrow 0$ for all vertices v except u

$d_w(v) \leftarrow \sum_{e \in E_w : v \in e} W(v)$

while $r(v)/d_w(v) \geq \varepsilon$ *for some node* $v \in V_w$ **do**

/* push operation */

$\rho \leftarrow r(v) - \frac{\varepsilon}{2} d_w(v)$; $\tilde{p}(v) \leftarrow \tilde{p}(v) + (1-\alpha)\rho$; $r(v) \leftarrow \frac{\varepsilon}{2} d_w(v)$

for *each* $x : (v, x) \in E_w$ **do** $r(x) \leftarrow r(x) + \frac{W(v,x)}{d_w(v)} \cdot \alpha\rho$

return $\tilde{p}$

However, the examples provided in the paper only employed simple motif, like

triangles.

- *EdWordle: Consistency-preserving Word Cloud Editing*

  EdWordle is a method for consistently editing word clouds.

Given a Wordle, use customized rigid body dynamics to move words close to each other. If a word is moved, the forces update the words accordingly. Empty spaces are moved by using a local version of the Wordle algorithm.

1.2 Vast Modification

System:

- Utility evaluation: add high level comparison, such as graph clustering, community detection and outlier detection (visualize in the graph view)
- Anonymity level: compute entropy-based metrics (display in the history view)
- Priority: add specific nodes by searching id

Writing:

1 Introduction (done)

- Explanation about k-anonymity
- Motivation
- Application

2 Related work (reorder subsections and add references about graph visualization)

- General graph visualization overview;
- Privacy-aware visualization;
- Privacy preservation for graphs;
- Evaluation

3 Task (done)

- Explanation about utility
- TR1: statistical plots or node-link visualization
- TR2: example of privacy vs utility
- TR3: explanation about "cost"

5 Visualization

- Details: choices of the design and metrics +refs

6 Evaluation

- Case study: focus on what the analyst discovered
- Expert review: more details about the views and the chosen algorithm.

7 Discussion

Limitation:

- Misperception of privacy protection
- Limited ability to resist complicated attacks
- Scalability and usability
- The conflicts generted by employing multiple protectors

1.3 Interview the Interns

2 Progress

Item	Deadline	Current progress	Remark
Vast modification	6.27	Summarized reviews and start to modify writing.	
Privacy program	10.31	Surveying.	Pausing.